

Escapy2.0 Engine Specification and User Guide

Генрих Тимур Домагальски

31.07.2017 Издание 1

Содержание

1	Начало работы	3
---	---------------	---

О движке

Escape2 это игровой движок написанный на java с использованием библиотек Dagger1, libGdx, box2d и Gson. Поскольку libGdx является лишь низкоуровневой оберткой над lwjgl - движок дает полноту простора в использовании openGL, в свою очередь Dagger делает код более модульным и масштабируемым.

1 | Начало работы

Вход производится похожим образом как в libGDX и lwjgl - в main с созданием инстанции **LwjglApplication**. Для этого создается объект **LwjglApplicationConfiguration** который загружается из json файла с помощью **EscapyDesktopConfigLoader**, о самих загрузчиках и механизме сериализации в движке более подробно потом.

```
import ...

public class DesktopLauncher {

    public static void main (String[] arg) throws Exception {

        LwjglApplicationConfiguration config = DesktopConfigLoaderBuilder.Default()
            .setLoadedClass(LwjglApplicationConfiguration.class)
            .setSerializedClass(SerializedDesktopConfig.class)
            .setPath(System.getProperty("user.dir") + separator + "Configuration.json")
            .setName("MainConfiguration")
            .build()
            .loadDesktopConfig();

        new LwjglApplication(new EscapyApplicationAdapter(MainEnvironment.class, new MainModule()), config);
    }
}
```

```
{
    "type": "EscapyConfiguration",
    "name": "MainConfiguration",

    "resizable": false,
    "vSync": true,
    "fullscreen": false,
    "forceExit": true,
    "useGL30": true,

    "scrWidth": 1920,
    "scrHeight": 1080,

    "fps": 25
}
```

При создании **LwjglApplication** в качестве аргумента передается **EscapyApplicationAdapter**, который в свою очередь в качестве аргумента принимает класс наследующийся от **EscapyGameContext** и varargs модулей Dagger'a.

```
import ...

public class MainEnvironment extends EscapyGameContext {

    private final EscapyScreen initialScreen;

    @Inject
    protected MainEnvironment(
        Collection<EscapyScreen> escapyScreens,
        EscapyScreen initialScreen,
        EscapyGameContextConfiguration contextConfiguration) {

        super(escapyScreens, contextConfiguration);
        this.initialScreen = initialScreen;
    }

    @Override
    protected EscapyScreen getInitialScreen() { return initialScreen; }

}
```

Подробнее о том как использовать модули Dagger'a можно прочитать на официальном сайте проекта (<http://square.github.io/dagger/>). **EscapyGameContext** имеет два конструктора, один из них как аргумент принимает инстанцию класса унаследованного от **EscapyGameContextConfiguration** - абстрактного класса предоставляющего конфигурацию проекта через методы которые можно перегрузить в случае необходимости.

```
import ...

public class MainConfiguration extends EscapyGameContextConfiguration {

    @Override
    public String getResourcesDir() {
        String local = System.getProperty("user.dir").replace((target: separator+"core"+separator+"assets", replacement: ""));
        return local + separator + "res";
    }

    @Override
    public String getConfigsFilePath() { return this.getResourcesDir() + separator + "configurations"; }

    @Override
    public void configurePropertyKeys(PropertyKeysStorage propertyKeyStorage) {
        propertyKeyStorage
            .addPropertyKey("BLEND_SHADERS_ROOT_DIR_PATH")
            .addPropertyValue(getResourcesDir() + separator + "shaders" + separator + "blend")
            .save();
    }

}
```

